AMENDMENTS TO THE CLAIMS

Please cancel claims 16-17 without prejudice. Kindly amend claims 1, 9, 15, 19, 23 and 24 as shown in the following listing of claims. Kindly add new claim 25 as shown in the following listing of claims. The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (currently amended) An apparatus in a pipelined microprocessor for invalidating a redundant entry for the same branch instruction in a set associative branch target address cache (BTAC), the apparatus comprising:

a way specifier, generated in a first pipeline stage, for specifying one of a plurality of ways of the BTAC for storing a target address of a branch instruction present in a cache line specified by an instruction cache fetch address missing in the BTAC;

a request, generated in a second pipeline stage, for requesting the BTAC to write a resolved target address of said branch instruction into said one of said plurality of ways specified by said way specifier in said first pipeline stage, wherein said second pipeline stage is subsequent to said first pipeline stage and said first and second pipeline stages are separated by at least three pipeline stages;

a status indicator, for indicating whether at least two ways of a set of the BTAC selected by an instruction cache fetch address contain a valid branch target address for a same branch instruction; and

control logic, coupled to said status indicator, for invalidating one of said at least two ways of said selected set if said status indicator indicates at least two ways of said selected set contain a valid branch target address for a same branch instruction.

2. (original) The apparatus of claim 1, further comprising:

a register, coupled to said control logic, for storing said instruction cache fetch address, for use by said control logic to invalidate said one of said at least two ways of said selected set.

3. (original) The apparatus of claim 1, wherein said selected set is selected by an index portion of said instruction cache fetch address.

4. (original) The apparatus of claim 1, wherein said control logic clears said status indicator after invalidating said one of said at least two ways of said selected set.

5. (original) The apparatus of claim 1, further comprising:

a register, coupled to said control logic, for storing data specifying said one of said at least two ways of said selected set to invalidate, for use in invalidating said one of said at least two ways of said selected set.

6. (original) The apparatus of claim 1, further comprising:

at least two valid signals, coupled to said control logic, each for indicating whether a respective one of said at least two ways of said selected set contain a valid branch target address.

7. (original) The apparatus of claim 1, further comprising:

at least two match signals, coupled to said control logic, each for indicating whether a tag portion of said instruction cache fetch address matches a tag stored in a respective one of said at least two ways of said selected set.

8. (original) The apparatus of claim 7, further comprising:

at least two comparators, coupled to said control logic, each for comparing said tag portion of said instruction cache fetch address with said tag stored in said respective one of said at least two ways of said selected set, and generating a respective one of said at least two match signals in response to said comparing.

9. (currently amended) An apparatus in a pipelined microprocessor for invalidating redundant entries for the same branch instruction in a branch target address cache (BTAC), comprising:

a way specifier, generated in a first pipeline stage, for specifying one of a plurality of ways of the BTAC for storing a target address of a branch instruction present in a cache line specified by an instruction cache fetch address missing in the BTAC;

a request, generated in a second pipeline stage, for requesting the BTAC to write a resolved target address of said branch instruction into said one of said plurality of ways specified by said way specifier in said first pipeline stage, wherein said second pipeline stage is subsequent to said first pipeline stage and said first and second pipeline stages are separated by at least three pipeline stages;

detection logic, for detecting a condition in which more than one valid way of a plurality of ways of a selected set of the BTAC are storing a target address for a same branch instruction; and

invalidation logic, coupled to said detection logic, for invalidating all but one of said more than one valid way of said selected set.

10. (original) The apparatus of claim 9, further comprising:

a register, coupled to said invalidation logic, for storing an instruction cache fetch address;

wherein said selected set is selected by an index portion of said instruction cache fetch address, wherein said invalidation logic invalidates said all but one of said more than one valid ways of said selected set using said instruction cache fetch address stored in said register.

11. (original) The apparatus of claim 10, wherein said detection logic is configured to receive a match signal for each of said plurality of ways of said selected set, said match signal specifying whether an address tag of said way matches an address tag portion of said instruction cache fetch address.

12. (original) The apparatus of claim 11, wherein said detection logic is further configured to receive a valid indicator for each of said plurality of ways of said selected set, said valid indicator specifying whether said way is valid.

13. (original) The apparatus of claim 12, further comprising:

a flag, coupled to said detection logic, for indicating whether more than one valid way of a plurality of ways of a selected set of the BTAC are storing a target address for a same branch instruction, wherein said detection logic sets said flag to a true value if for more than one of said plurality of ways said match signal and said valid indicator are true.

14. (original) The apparatus of claim 13, wherein said invalidation logic invalidates said one of said more than one valid ways of said selected set if said flag is true.

15. (currently amended) A pipelined microprocessor, comprising:

an instruction cache, having an address input for receiving an address to select a line including a branch instruction;

a branch target address cache (BTAC), coupled to said instruction cache, for generating a plurality of indicators in response to said address, each of said plurality of indicators indicating whether a corresponding way in a set of said BTAC selected by said address is storing a valid target address of said branch instruction; ~~and~~

logic, coupled to said BTAC, configured to invalidate one or more of said plurality of ways of said selected set if said plurality of indicators indicates two or more of said plurality of ways is storing a valid target address of said branch instruction;

a first pipeline stage, in which said BTAC indicates a miss of said address therein, and in which said BTAC specifies one of said plurality of ways for storing said target address; and

a second pipeline stage, subsequent to said first pipeline stage, which requests said BTAC to write a resolved target address of said branch instruction into said one of said plurality of ways specified by said BTAC in said first pipeline stage, wherein said first and second pipeline stages are separated by at least three pipeline stages.

16-17. (canceled)

18. (currently amended) The microprocessor of ~~claim 16~~ claim 15, wherein it is possible for a subsequent fetch of said branch instruction from said instruction cache to reach said first stage prior to a previous fetch of said branch instruction reaching said second stage, such that said selected set of said BTAC stores a valid target address of said branch instruction in two or more of said plurality of ways.

19. (currently amended) A method for invalidating redundant entries in a set-associative branch target address cache (BTAC) for the same branch instruction, the method comprising:

determining whether a tag of more than one way of a set of the BTAC selected by an index portion of an instruction cache fetch address matches a tag portion of the instruction cache fetch address and is valid; ~~and~~

invalidating all but one way of the selected set, if more than one way of the selected set is valid and matching;

indicating, in a first pipeline stage, a miss of said address in said BTAC and specifying one of said plurality of ways for storing a target address of a branch instruction included in a cache line selected by said address in an instruction cache; and

requesting said BTAC, in a second pipeline stage, to write a resolved target address of said branch instruction into said one of said plurality of ways specified by said BTAC in said first pipeline stage, wherein said second pipeline stage is subsequent to said first pipeline stage and said first and second pipeline stages are separated by at least three pipeline stages.

20. (original) The method of claim 19, further comprising:

storing an indication that a tag of more than one way of a set of the BTAC selected by an index portion of an instruction cache fetch address matches a tag portion of the instruction cache fetch address and is valid in response to said determining.

21. (original) The method of claim 19, further comprising:

storing said instruction cache fetch address, in response to said determining.

22. (original) The method of claim 19, further comprising:

storing an indication of said all but one way of the selected set that are to be invalidated, in response to said determining.

23. (currently amended) A method for invalidating a redundant entry for the same branch instruction in the same set of an N-way set associative branch target address cache (BTAC), the method comprising:

selecting an N-way set in the BTAC with a lower portion of an instruction fetch address;

comparing N address tags of N corresponding ways of said N-way set with an upper portion of said instruction fetch address;

determining whether two or more of said N address tags match said upper portion and are valid; ~~and~~

invalidating, if two or more of said N address tags match said upper portion and are valid, one or more of said N ways corresponding to said two or more of said valid N address tags matching said upper portion;

indicating, in a first pipeline stage, a miss of said address in said BTAC and specifying one of said N ways for storing a target address of a branch instruction included in a cache line selected by said address in an instruction cache; and

requesting said BTAC, in a second pipeline stage, to write a resolved target address of said branch instruction into said one of said N ways specified by said BTAC in said first pipeline stage, wherein said second pipeline stage is subsequent to said first pipeline stage and said first and second pipeline stages are separated by at least three pipeline stages.

24. (currently amended) A computer ~~data signal~~ program embodied ~~in~~ on a ~~transmission~~ computer-readable medium, comprising:

computer-readable program code for providing a pipelined microprocessor, said program code comprising:

first program code for providing an instruction cache, having an address input for receiving an address to select a line including a branch instruction;

second program code for providing a branch target address cache (BTAC), coupled to said instruction cache, for generating a plurality of indicators in response to said address, each of said plurality of indicators indicating whether a corresponding way in a set of said BTAC selected by said address is storing a valid target address of said branch instruction; ~~and~~

third program code for providing logic, coupled to said BTAC, configured to invalidate one or more of said plurality of ways of said selected set if said plurality of indicators indicates two or more of said plurality of ways is storing a valid target address of said branch instruction;

fourth program code for providing a first pipeline stage, in which said BTAC indicates a miss of said address therein, and in which said BTAC specifies one of said plurality of ways for storing said target address; and

fifth program code for providing a second pipeline stage, subsequent to said first pipeline stage, which requests said BTAC to write a resolved target address of said branch instruction into said one of said plurality of ways specified by said BTAC in said first pipeline stage, wherein said first and second pipeline stages are separated by at least three pipeline stages.

25. (new) The computer program embodied on a computer-readable medium of claim 24, wherein it is possible for a subsequent fetch of said branch instruction from said instruction cache to reach said first stage prior to a previous fetch of said branch instruction reaching said second stage, such that said selected set of said BTAC stores a valid target address of said branch instruction in two or more of said plurality of ways.